



TAMPEREEN
AMMATTIKORKEAKOULU
LIIKETALOUS

TUTKINTOTYÖRAPORTTI

Tulostuspalvelu tekstiiliteollisuudelle

Klaus Laikola

Tietojenkäsittelyn koulutusohjelma
Toukokuu 2005
Työn ohjaaja: Harri Hakonen

TAMPERE 2005



Tekijä(t): Laikola Klaus
Koulutusohjelma(t): Tietojenkäsittelyn koulutusohjelma
Tutkintotyön nimi: Tulostuspalvelu tekstiiliteollisuudelle

Title in English: Printing service for textile industry

**Työn valmistumis-
kuukausi ja -vuosi:** Toukokuu 2005

Työn ohjaaja: Hakonen Harri

Sivumäärä: 48

TIIVISTELMÄ

Opinnäytetyöni toimeksiantaja on tekstiilialan yritys. Yrityksessä yhdellä osa-alueella tietoa tallennettiin perinteisesti kynällä ja paperilla. Joissain yrityksissä perinteisessä tavassa pysyttäytyminen voi olla kätevintäkin, kun toiminta on vähäistä, eikä tallennettavaa tietoa ole paljon. Usein toimintatapoihin halutaan saada muutos yritystoiminnan laajentuessa. Toimeksiantajayrityksen toiminnan kasvaessa sen tietojenkäsittelyä haluttiin kehittää nykyaikaisemmaksi.

Tavoitteenani oli tehdä tietokanta ja ohjelmoida ohjelma, joka on yhteydessä tekemääni tietokantaan. Ohjelman avulla tietokannasta pystytään hakemaan ja poistamaan tietoa, lisäksi sinne voidaan tallentaa uusia tietoja. Ohjelmalla pystytään myös tulostamaan pakkalappuja lähetettäviin tekstiileihin.

Muutoksen jälkeen tiedon tallentaminen on varmempaa ja nopeampaa kuin nyt. Työ mahdollistaa tiedon helpomman jälkikäsitteilyn tämän hetken käytäntöön verrattuna.

Raportissani yrityksen ja ohjelman toimintaympäristön esittelyn jälkeen esittelen työssäni käytettävät menetelmät. Ohjelmointikielenä käytin Visual Basic.NETiä. Tietokannaksi valitsin Microsoft Access:n. Suurin osa kirjallisesta raportoinnista pitää sisällään työn, eli ohjelman ja tietokannan toimintaa, joka on esitelty kuvien ja koodiesimerkkien avulla.

Avainsanat: Access Bluetooth Visual Basic.NET Relaatiotietokanta Tulostus

Sisällysluettelo:

1 Johdanto.....	6
2 Taustaa.....	7
2.1 Aiheen valinta.....	7
2.2 Työn toimeksiantaja	7
2.3 Tilanteen selvitys.....	7
2.4 Koneen esittely	8
3 Työn tavoitteet ja käytettävät menetelmät.....	9
3.1 Ohjelman ja tietokannan tavoitteet.....	9
3.2 Käytettävät menetelmät	10
4 Visual Basic.NET	11
4.1 Käyttöliittymäsuunnittelu	11
4.2 Tietokantasovellukset	11
5 Access lyhyesti	13
5.1 Tärkeimmät ominaisuudet	13
5.2 Accessin käyttö.....	13
6 Yleistä Relaatietietokannoista	15
6.1 Taulukot.....	15
6.2 Avainkentät ja relaatiot.....	16
7 Bluetooth	18
7.1 Standardit.....	18
7.2 Bluetoothin käyttäminen	18
7.3 Tiedonsiirto.....	19
7.4 Verkko	20
7.5 Bluetooth teollisuudessa.....	21
8 Työn sisältö	22
8.1 Aloitussivu.....	22
8.2 Ohjelman pääsivu	24
8.3 Väri- ja laatu tietojen lisäys ja poistaminen	39
9 Tietokanta	45
9.1 Tietokannan tietotyypit ja primary key:t	45
9.2 Tietokannan yhteydet	46
10 Pohdinta	47
Lähteet	48

Sanastoa

ACL

(Asynchronous Connection – Less), Bluetoothin tiedonsiirtomenetelmä, joka on tarkoitettu datan siirtoon.

Erä

Erällä tarkoitetaan yhden tai useamman tekstiilipakan kokonaisuutta, joka värjätään tai käsitellään samalla kertaa.

ISM-kaista

Taajuusalue, joka kuuluu maailmanlaajuisesti vapaasti käytettäviin ja ilmaisiin kaupan, teollisuuden ja tieteen käyttöön varattuihin ISM-taajuuksiin (Industrial, Scientific, Medical).

Kutistuma

Kutistumalla tarkoitetaan sitä, kun valmis tuote esim. paita laitetaan pesuun, kuinka paljon se saa kutistua pituus- ja leveyssuunnassa.

Laatu

Laatu yksilöi tekstiilin esim. college, single, resori, RA64065.

Master

Picoverkossa voi olla aktiivisena yksi isäntä.

Neliöpaino

Neliöpaino on tekstiilin paino g/m². Raamin asetuksia säätämällä, tekstiiliä venyttämällä pituus- tai leveyssuunnassa saadaan haluttu neliöpaino.

Pakka

Pakalla tarkoitetaan yhtä rullaa tekstiiliä.

Pakkalappu

Pakkalappu on valmiiden tekstiilipakkojen päälle liimattava lappu, jossa on tiedot asiakkaasta ja tekstiilistä sekä päivämäärä ja värjäämön nimi.

Raami

Kone, jolla tekstiilit kuivataan ja rullataan.

RAD

Rapid Application Development. RAD:lla nopeutetaan prototyypiohjelmien laatimista sekä käyttöliittymäsuunnittelua.

RDBMS

(Relation Database Management System)Relaatiotietokantojen hallintajärjestelmä.

Resepti

Reseptissä on esikäsittely- ja värjäysohjeet.

SCO

(Synchronous Connection – Oriented), Bluetoothin tiedonsiirtomenetelmä, joka on tarkoitettu äänen siirtoon.

SIG

(Special Interest Group) on ryhmä, joka pitää sisällään yli 2000 tekniikasta kiinnostunutta yritystä. Ryhmä pyrkii kehittämään tekniikkaa ja sen markkinoille tuontia.

Slave

Picoverkossa voi olla seitsemän orjalaitetta.

1 Johdanto

Opinnäytetyöni aiheena on ”Tulostuspalvelu tekstiiliteollisuudelle”. Työni tavoitteena oli kehittää yrityksen tietojenkäsittelyä. Toimeksiantajayrityksessä yhdellä osa-alueella tiedon tallentamiseen käytettiin vielä kynää ja paperia. Toiminnan kasvaessa tilanteeseen haluttiin saada muutos. Ohjelmoin tulostusohjelman sekä tein tietokannan, jota ohjelma käyttää tiedon tallentamiseen. Ohjelman myötä tiedon tallentaminen on helpompaa ja luotettavaa. Tallentaminen tietokantaan tekee tiedon jälkikäsittelyn mahdolliseksi.

Ohjelman toisena tavoitteena oli tulostaa valmiita pakkalappuja. Pakkalappuja tarvitaan tekstiilien kuivauksessa ja rullauksessa käytettävällä Raami-koneella. Pakkalaput liimataan asiakkaille lähetettäviin tekstiileihin. Pakkalapulle tulostuvat tiedot asiakkaasta ja tekstiilistä. Tiedot valitaan ohjelmasta ja ne tallennetaan tietokantaan.

Raporttini teoreettisessa viitekehyksessä käsittelen ohjelmassa käytettäviä menetelmiä. Ohjelmointikielenä käytin Visual Basic.NETiä. Tietokannaksi valitsin Microsoftin Access 2000:n. Esittelen myös relaatiotietokantojen perusteita sekä tulostuksessa käytettävää Bluetooth-teknologiaa. Raportissani esittelen ohjelman toteutuksen kuvien ja koodiesimerkkien avulla.

2 Taustaa

2.1 Aiheen valinta

Opinnäytetyöni aihe on Tulostuspalvelu tekstiiliteollisuudelle. Olen työskennellyt yrityksessä muutaman vuoden, sekä opiskelun ohessa että ennen kouluun lähtemistä. Tällä hetkellä olen vakituksessa työsuhteessa yrityksessä. Idea tähän aiheeseen syntyi, koska tiesin hyvin mikä yrityksen toiminnassa vaati kehittämistä. Aikaisempi tiedon tallentaminen tapahtui kynällä ja paperilla. Yrityksen toiminnan kasvaessa tähän piti saada muutos, koska tallennettavaa tietoa oli entistä enemmän. Yrityksessä kiinnostuttiin kehittämisideasta ja se päätettiin toteuttaa.

2.2 Työn toimeksiantaja

Työn toimeksiantaja on Värjäämö Laikola & Co. Yritys on tekstiilivärjäämö, joka työllistää n. 10 henkilöä. Toimintaan kuuluu tekstiilin värjäys-, valkaisu- ja viimeistelypalvelut. Yritys tuo myös maahan tekstiiliä EU-alueelta. Vuosittain yrityksessä värjätään ja käsitellään tekstiilejä n. 200 000 - 250 000 kg:a. Toimenkuvaani yrityksessä kuuluvat tuotannon työnteon lisäksi yrityksen mikrotuki.

2.3 Tilanteen selvitys

Ennen tekstiilin käsittelyä, asiakas tuo itse tai lähettää kuljetuspalvelun kautta tekstiilin värjäämölle. Asiakas voi myös tilata yrityksen maahantuomista tekstiileistä tarvitsemansa. Useille asiakkaille yrityksellä on pieni varasto.

Tilauksen jälkeen tehdään erä, joka sisältää asiakkaan tilaamat tekstiilit. Erällä tarkoitetaan sitä määrää tekstiiliä, joka värjätään tai käsitellään yhdessä osassa. Yksi erä pitää sisällään 1–20 pakkaa. Pakalla tarkoitetaan yhtä rullaa tekstiiliä. Tekstiilit laitetaan koneeseen ja tehdään resepti. Resepti on ohje, jonka mukaan tekstiilit käsitellään ja värjätään. Värjäyksen jälkeen tekstiilit laskostetaan ja siirretään Raamille. Raami on kone, jolla tekstiilit kuivataan ja rullataan. Opinnäytetyönäni tekemää tulostuspalvelua käytetään Raamilla.

Kuivauksen ja rullauksen jälkeen Raamin käyttäjä tarvitsee pakkalappuja erälle. Pakkalappu on valmiiden tekstiilipakkojen päälle liimattava lappu, jossa on tiedot asiakkaasta ja tekstiilistä sekä päivämäärä ja värjäämön nimi.

Tällä hetkellä pakkalaput tulostetaan tavalliselle paperiarkille ja tiedot kirjoitetaan siihen käsin. Tekstiilin vastaanottaja katsoo mittalaitteesta pakan pituuden, kun pakka katkaistaan.

Koko erän tiedot, joka sisältää useita pakkoja, pitää myös kirjata. Kun tekstiilit on kuivattu ja rullattu, ne paketoidaan. Valmiit, paketoitut pakat lähetetään asiakkaalle. Tulostuspalvelun myötä, ohjelma tulostaa valmiita pakkalappuja, jossa tiedot ovat valmiina. Ohjelma tallentaa koko erän tiedot tietokantaan.

2.4 Koneen esittely

Raami on kone, jolla tekstiilit kuivataan ja rullataan. Kone on kokonaisuudessaan 25m pitkä ja 4m leveä. Kuivauksen yhteydessä koneella säädetään tekstiilille kutistumat sekä neliöpainot. Kutistumalla tarkoitetaan sitä, kun valmis tuote esim. paita laitetaan pesuun, kuinka paljon se saa kutistua pituus- ja leveyssuunnassa. Asiakas antaa tekstiileille raja-arvot, joiden välissä kutistuman tulee olla. Neliöpaino on tekstiilin paino g/m². Raamin asetuksia säätämällä, tekstiiliä venyttämällä pituus- tai leveyssuunnassa saadaan haluttu neliöpaino.

Raamilla työskentelee kaksi henkilöä. Toinen heistä on pääkäyttäjä, joka käyttää konetta. Toinen henkilö vastaanottaa kuivattuja ja rullattuja pakkoja, paketoit ne ja liimaa pakkalaput.



Kuva 2.1 Raami

3 Työn tavoitteet ja käytettävät menetelmät

Opinnäytetyöni tavoitteena oli tehdä ohjelma ja tietokanta. Ohjelmointikieleksi valitsin Visual Basic.NET:n. Tietokantana käytin Microsoft Access:ia. Tulostimen ja ohjelmaa käyttävän tietokoneen väliseksi tiedonsiirtomenetelmäksi valitsin Bluetooth:n.

3.1 Ohjelman ja tietokannan tavoitteet

Ohjelman tavoitteena on tulostaa valmiita pakkalappuja, jotka liimataan lähetettäviin tekstiilipakkoihin. Pakkalappu on valmiiden tekstiilipakkojen päälle liimattava lappu, jossa on tiedot asiakkaasta ja tekstiilistä sekä päivämäärä ja värjäämön nimi. Tiedot koko erästä tallennetaan tietokantaan.

Ohjelmaa käytetään Raamilla kuivattujen ja rullattujen, pakoista koostuvien erien tietojen tallentamiseen tietokantaan. Ohjelma tallentaa tietokantaan koko erästä seuraavat tiedot:

- asiakkaan nimi
- laadun nimi
- laadun numero
- värin nimi
- värin numero
- erän pituus
- pvm.

Sanaa laatu, käytetään tekstiilin yksilöimiseen. Laadun nimi on esim. college, single, resori, RA64065. Laadun numero on laadun nimelle numeerinen vastine. Nämä tiedot ml. värit, ovat yrityksen ja asiakkaan välisiä, tekstiilin ominaisuuksia kuvaavia merkintöjä.

Ohjelma tulostaa pakkalappuja. Pakkalappu sisältää tietokantaan tallentuvat tiedot lukuunottamatta erän pituutta. Tässä kohtaa pakkalapussa lukee pakanpituus. Ohjelman käyttäjän on voitava valita seuraavat tiedot: asiakas, laadun nimi, laadun numero, värin nimi ja värin numero. Nämä tiedot ohjelma hakee tietokannasta.

Muita ohjelmalle asetettuja vaatimuksia ovat tallennettujen erien selailu näytöllä sekä tallennetun erän poistaminen tietokannasta. Ohjelman pitää myös laskea eränumerot jokaiselle erälle. Eränumerot uusille asiakkaille lähtevät 1:stä ja kasvavat aina yhden seuraavaan erään. Vanhan käytännön mukaan lasketut eränumerot on voitava siirtää suoraan ohjelman tietokantaan.

Ohjelmassa on oltava kohta, josta tulostetaan tyhjiä pakkalappuja. Tyhjiä pakkalappuja tarvitaan sellaisille tekstiileille, joiden pituuksia ei mitata.

Tämän pääsivun lisäksi ohjelmassa on aloitussivu, josta käyttäjä kirjautuu ohjelmaan. Ohjelmassa tulee olla sivu, jossa laatu- ja väritietoja voidaan tallentaa tietokantaan, sekä poistaa tietokannasta.

3.2 Käytettävät menetelmät

Päätin etukäteen mitä ohjelmia tulen käyttämään. Ohjelmana käytin Visual Basic.Net:iä. Perustelu tämän ohjelman käyttämiseen oli sen helppous. Helppoudella en tarkoita sitä, että sillä ohjelmointi olisi helppoa, vaan vertasin sitä muihin ohjelmointikieliin. Tämä päättyi ainoaksi vaihtoehdoksi yrittää tehdä ohjelmaa. Aikaisempaa kokemusta minulla oli tästä vähän ja muista ohjelmointikielistä vielä vähemmän, joten valinta oli selvä.

Tietokannan vaihtoehtoina olivat MySQL ja Access. Aluksi minulla oli mielikuva siitä, että MySQL olisi luotettavampi tietokanta. Kokemuksia ei minulla ollut ja toisaalta olin myös lukenut, että Access olisi virallisempi tietokanta Visual Basicille. Tietokannaksi valitsin Microsoft Accessin. Olin käyttänyt tätä vähän ennenkin ja kokemus muiden Microsoftin tuotteiden käytöstä auttoi valinnassa.

Ohjelman käyttämäksi tiedonsiirtotekniikaksi valitsin Bluetooth:n. Valinnan tein mielenkiinnon pohjalta ja tiesin sen soveltuvan tulostimiin. Bluetoothin valintaan vaikutti myös sen langattomuus, jonka avulla voitaisiin saada lisäarvoa tulevaisuudessa esim. laitteiden helpolla siirrettävyydellä.

4 Visual Basic.NET

Visual Basic on sovelluskehitin, joka perustuu Basic-kieleen. Se on osa Visual-Studio kokonaisuutta. Visual-Studio sisältää ohjelmistotuotantoon tarkoitettuja ohjelmia. Visual Basic on RAD-sovelluskehitin. RAD on lyhennys sanoista Rapid Application Development. RAD:lla nopeutetaan prototyyppiohjelmien laatimista sekä käyttöliittymäsuunnittelua. Visual Basic ei ole täydellinen oliokieli, vaikka se tukeekin oliopohjaista suunnittelu- ja toteutusmallia.(Vitikka 2000.)

”Microsoft Visual Basic.NET on Visual Basic-kehitysjärjestelmän uusiin ja tehokkain versio. Se on suunniteltu Microsoft .NET Framework-ympäristön ensiluokkaiseksi kieleksi, ja se tarjoaa helpoimman ja tuottavimman tavan uusille kehittäjille sekä Visual Basic-ohjelmoijille luoda nopeasti Windows-pohjaisia sovelluksia, seuraavan sukupolven XML-Web-palveluita, verkkotietokoneiden Web-sovelluksia ja langattomien laitteiden ohjelmistoja.” (Microsoft Corporation 2004-Tuotetietokeskus.)

Visual Basic soveltuu sekä kehittyneemmille, että aloitteleville ohjelmoijille. Sen matalan oppimiskynnyksen myötä se on levinnyt laajasti varsinkin Windows-sovellusten maailmassa. Visual Basic soveltuu käyttöliittymäsuunnitteluun sekä tietokantasovellusten laatimiseen. (Vitikka 2000.)

4.1 Käyttöliittymäsuunnittelu

Käyttöliittymäsuunnittelu voi sisältää useita prototyyppisovelluksia. Uusien käyttöliittymien suunnittelu voi olla työlästä ja aikaa vievää. Tähän tarkoitukseen RAD-sovelluskehitin tuo mukanaan paljon etuja, sen nopean käytettävyyden vuoksi. Käyttöliittymän pystyy tekemään pienellä määrällä itse kirjoitettua koodia. Nopeus perustuu sovelluskehittimen valmiskomponentteihin. Valmiita komponentteja, kuten tekstikenttiä sekä nappeja yms. saadaan valmiina suoraan valikosta.(Vitikka 2000.)

4.2 Tietokantasovellukset

Tietokantasovellusten käyttöä on pyritty helpottamaan monissa sovelluskehittimissä. Myös Visual Basicissa on tämä otettu huomioon. Visual Basicin peruskomponentteja pystytään lähes poikkeuksetta liittämään suoraan tietokantaan ja siten saamaan tietokannasta suoraan tieto. Visual Basic soveltuu tietokantasovelluksiin hyvin.(Vitikka 2000.)

Visual Basic.NET järjestelmävaatimukset:

- Suoritin
PC-tietokone min. 450 MHz:n Pentium 2-suoritin
- Muisti
Windows NT Workstation 4.0: 64 megatavua; Windows 2000 Professional: 96 megatavua; Windows 2000 Server: 192 megatavua; Windows XP Professional: 160 megatavua
- Kiintolevy
2 500 megatavua kiintolevytilaa, josta vähintään 500 megatavua on käyttöjärjestelmäasemassa
- Asema
CD- tai DVD-asema
- Näyttö
Super VGA-näyttö (vähintään 800 x 600 kuvapisteen tarkkuus ja 256 väriä).
- Käyttöjärjestelmä
Windows 2000®, Windows XP ja Windows NT 4.0

Microsoft Mouse-hiiri tai muu yhteensopiva osoitinlaite

(Microsoft Corporation 2004–Tuotetietokeskus.)

5 Access lyhyesti

Microsoft Access tietokantaohjelma on relaatiotietokantojen hallintajärjestelmä (RDBMS). Se on osa Office kokonaisuutta. Työssäni käytettävä Access 2000 on viides versio ohjelmasta. Hallintajärjestelmällä tarkoitetaan sitä, että se antaa käyttäjälle täyden kontrollin tiedon käsittelyyn. Käyttäjä pystyy vaikuttamaan siihen, kuinka hän haluaa käsitellä tietoa esim. tiedon määrittely, tiedon muokkaaminen ja tiedon hallinta.(Viescas 2000: 3 - 6.) Accessin avulla käyttäjän on helppo suunnitella ja rakentaa tietokantasovelluksia ilman ohjelmointitaitoja.(Viescas 2000: 14.)

5.1 Tärkeimmät ominaisuudet

Kirjassa (Microsoft Access 2000), on pitkä lista Accessin tärkeistä ominaisuuksista. Monet ominaisuudet merkitsevät erilaista hyötyä käyttäjälleen. Helppokäyttötoiminnot ovat luonnollisesti iso hyöty sellaiselle henkilölle, jolla ei ole kokemusta ohjelman käytöstä. Näiden avulla voi luoda taulukoita ja kyselyitä sekä muokata haluamiasi lomakkeita ja raportteja. Ohjattujen toimintojen ominaisuuden avulla on mahdollista parantaa tietokannan suorituskykyä.(Viescas 2000: 19.)

5.2 Accessin käyttö

Accessia voi käyttää, joko yksittäisenä relaatiotietokannan hallintajärjestelmänä tai jaettuna asiakas/ palvelin järjestelmänä verkossa. (Viescas 2000: 12.)

Accessin käyttötavat, sekä käyttäjät voivat vaihdella paljon. Kirjan (Microsoft Access 2000) esimerkissä on mainittu pienet yritykset, joissa sitä voidaan käyttää laskutuksessa, tilausten syötössä, sekä asiakasrekisterissä. Suuret yritykset, joissa tietokantaa voidaan käyttää työryhmäsovelluksissa sekä mm. intranet-julkaisuissa. Muita käyttäjäryhmiä esimerkissä ovat konsultit, myynti- ja markkinointiyritykset, sekä henkilökohtainen käyttö.

Microsoft Access 2000:n järjestelmävaatimukset

- vähintään 75 MHz:n Pentium-suorittimella varustettu PC
- Microsoft Windows® 95 -käyttöjärjestelmä tai uudempi tai Microsoft Windows NT® Workstation 4.0-käyttöjärjestelmä ja Service Pack 3 tai uudempi
- Windows 95:n tai Windows 98:n kanssa: - 16 megatavua muistia käyttöjärjestelmää varten ja 8 megatavua muistia Accessia varten
- Windows NT Workstationin kanssa: - 32 megatavua muistia käyttöjärjestelmää varten ja 8 megatavua muistia Accessia varten
- 161 megatavua vapaata levytilaa (Tämä määrä riittää tyypilliseen asennukseen. Levytilan tarve vaihtelee määritysten mukaan. Mukautetun asennuksen aikana tehdyt valinnat voivat vaatia joko enemmän tai vähemmän levytilaa kuin tässä on ilmoitettu.)
- CD-asema
- vähintään VGA-näyttö (Super VGA-näyttöä suositellaan)
- Microsoft Mouse-hiiri, Microsoft IntelliMouse-hiiri tai yhteensopiva osoitinlaite. (Microsoft Oy 1999, Office 2000 – Microsoft Access.)

6 Yleistä Relaatiotietokannoista

Relaatiotietokannat koostuvat monista eri taulukoista. Relaatiotietokannassa (Relation Database) tiedot ryhmitellään eri taulukoihin. Relaatioiden tarkoitus on mahdollistaa tietojen käsittely monesta eri taulukosta samaan aikaan. Suurin etu relaatiotietokannoista saavutetaan silloin, kun useaan taulukkoon lisätään tai poistetaan tietoa samaan aikaan. Silloin muutokset tapahtuvat heti ja ne saadaan saman tien käyttöön.

Jos verrataan relaatio- ja ei-relaatiotietokantaa, suurin ero on tiedon tallennuksessa. Ei-relaatiotietokannassa sama tieto on tallennettu useaan eri paikkaan ja siitä johtuen myös muutokset on myös tehtävä useaan eri paikkaan. (Virenius. 1997: 10.)

Relaatiotietokannan hallintajärjestelmä (Relational Database Management System) on relaatiotietokannan käyttämiseen tarkoitettu erillinen ohjelma. Microsoftilla näitä on kaksi: Access ja Microsoft SQL. Microsoft SQL on monen samanaikaisen käyttäjän ympäristöön tarkoitettu. Nämä relaatiotietokantaohjelmat tarjoavat työkalun tietokantojen hallintaan. Näiden avulla voidaan tietokantoja perustaa sekä muokata ja hallita niiden tietoja. (Sainio. 2002: 6.)

6.1 Taulukot

Tietokantojen sisältämä tieto on tallennettu taulukoihin (Table). Taulukolla tarkoitetaan paikkaa, jonne on tallennettu yhteen tiettyyn aiheeseen kuuluvat tiedot. Taulukko sisältää rivejä, jotka ovat tietueita (Record). Tämä tietue pitää sisällään yhden esim. tuotteen tietoja. Taulukot sisältävät myös kenttiä (Field). Kentät ovat pystysuoria sarakkeita, joissa tieto on yhdenmukaista. (Sainio. 2002: 6.)

Taulukko(Table)

Laatu

laatu_id	laatu_nimi	asiakas	laatu_nro
33	college	Asiakas1	700
34	Interlock		9065
35	resori	Asiakas3	7288

Kuva 6.1 Taulukko

Kuvassa 6.1 on taulukko Laatu. Laadun yksi kenttä on laatu_nimi, joka sisältää erilaisten laatuojen nimiä. Taulukon tietueita ovat yhden rivin tiedot, joka sisältää yhdelle laadulle kuuluvat tiedot. Taulukoissa on myös soluja (Cell).

Solu on kentän ja tietueen leikkauskohta. Soluihin on tallennettu aina yhden tiedon arvo. (Sainio. 2002: 6) Esim. Taulukossa Laatu sijaitsevan Asiakas kentän arvo on Asiakas1. Asiakas kenttä voi olla myös tyhjä. Silloin sen ominaisuudeksi pitää laittaa (Null), joka sallii tyhjän arvon.

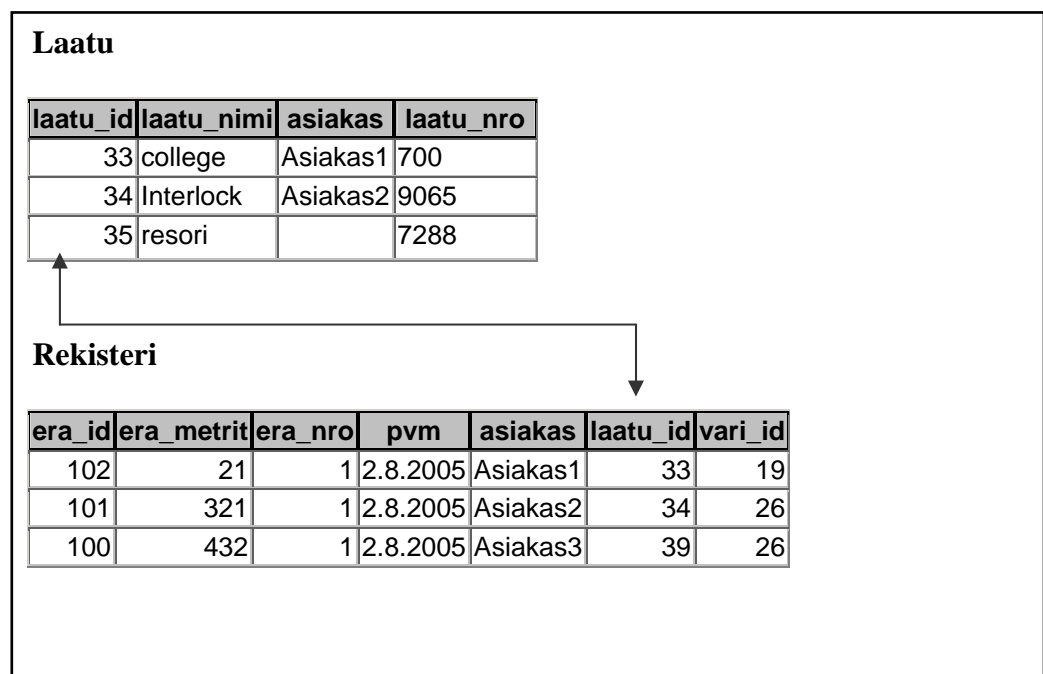
Access tietokannassa ja yleisesti kirjallisuudessa olevat termit poikkeavat. Tässä ovat esitelty niiden vastaavat termit:

Access	Yleiset kirjallisuudet tietokannoista
Taulukko	Taulu
Tietue	Rivi
Kenttä	Sarake
Solu	Kenttä

Kuva 6.2

6.2 Avainkentät ja relaatiot

Jokaisessa taulukossa pitää olla Perusavain (Primary Key). Tämä tieto erottaa tietueet toisistaan. Relaatiot (Relationship) yhdistävät eri taulukot toisiinsa. Yhteyden muodostaminen tapahtuu saman tiedon sisältävien kenttien avulla. (Sainio 2002: 7.)



Kuva 6.3 Taulukot

Kuvassa 6.3 esitetään Rekisteri- ja Laatu taulukoiden yhdistäminen. Molemmista taulukoista on kenttä Laatu_id. Rekisteri taulukossa se kertoo mikä Laatu_id on kyseessä. Laatu taulukossa sen avulla saadaan tarkempaa tietoa laadusta. Laatu kentän perusavain on Laatu_id, joka yksilöi laadut. Perusavain ei voi sisältää tyhjää arvoa.

Viiteavain (Foreign Key) on linkki toisen taulun perusavaimen. Tämän avulla voidaan taulukoita yhdistää, jolloin voidaan kerätä tarkempia tietoja toisesta taulusta sijaitsevista tiedoista. Kuvassa 6.3 viiteavaimena taulukkoon Laatu toimii taulukon Rekisteri kenttä laatu_id. (Sainio 2002: 7.)

7 Bluetooth

Bluetooth tiedonsiirtoteknologia on suunniteltu lyhyille matkoille korvaamaan kaapelit. Suunnittelun lähtökohta on ollut laitteen yksinkertaisuus.

Muita tavoitteita ovat:

- ympäristöolojen sietokyky,
- alhainen tehonkulutus sekä
- halpa hinta.

Bluetoothin radiolinkki käyttää vapaata 2.4 gigahertsin taajuudella toimivaa kaistaa. Kaista toimii ympäri maapallon, muutamia maita lukuunottamatta. (Karasti 2000.)

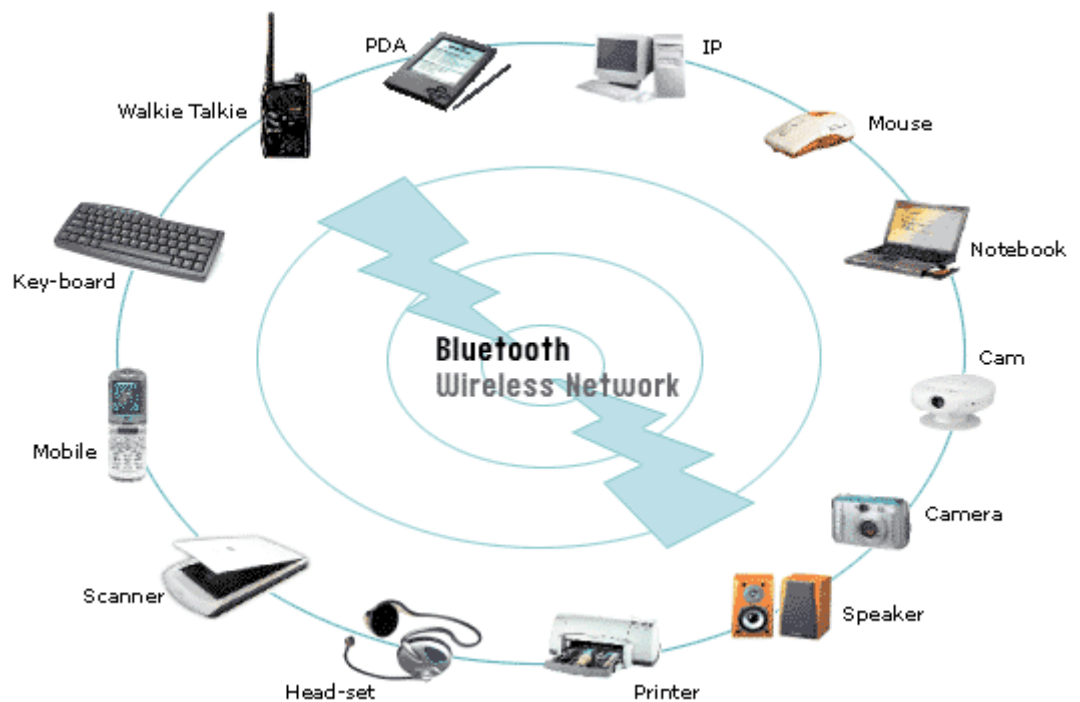
Bluetoothin avulla on mahdollista saavuttaa jopa noin 100 m:n kantavuus. Tämä kasvattaa laitteen kokoa ja hintaa. Bluetooth on samanlainen kuin mikä tahansa muu tiedonsiirtotekniikka. Siinä on lähetin ja vastaanotin. Suurimpana erona tavalliseen tiedonsiirtoon voidaan pitää sitä, että se kommunikoi radioaaltoja pitkin. (Kontio, Tervo... 2002:68.)

7.1 Standardit

Ensimmäinen standardi oli versio 1.0, joka julkaistiin vuonna 1999. Tämä oli oikeastaan ensimmäinen versio specificaatiosta. Tästä seuraavia olivat 1.0b ja 1.1 vuonna 2001. Standardi sisältää kaksi päädokumenttia: ydindokumentti ja profiilidokumentti. Profiilidokumentti sisältää malleja ja tilanteita, joissa Bluetoothia käytetään. Ydindokumentti pitää sisällään suunnittelun ohjeet. Koko standardin pituus on 1500 sivua. (Kontio, Tervo... 2002:68.) Bluetooth (SIG) Special Interest Group on ryhmä, joka pitää sisällään yli 2000 tekniikasta kiinnostunutta yritystä. Ryhmä pyrkii kehittämään tekniikkaa ja sen markkinoille tuontia. (Kontio, Tervo... 2002:67.) Nyt SIG:n toimesta on määritetty uusi versio 2.0. (Hänninen 2004.)

7.2 Bluetoothin käyttäminen

Käyttömahdollisuuksia Bluetoothilla on monia. Esim. puheen- ja tiedonsiirrot sekä automaattiset push-toiminnot. Bluetoothissa on yleisprofiili (General Access Profile). Tämä määrittelee yhteiset toimintaperiaatteet Bluetooth-laitteille. Tästä yleisprofiilista periytyvät muut profiilit. Esimerkkinä Bluetooth-laite voi toimia puhelimenä puhelinverkossa tai puhelimen luurina.



Kuva 7.1 Bluetoothin käyttökohteet (cool.tpage.com)

Push-ominaisuus, ”spontaani työntäminen”, on yksi Bluetoothin ehkä eniten suunnitelluista ominaisuuksista. Suurin osa siitä on mainostamista esim. kaupan tarjouksista. Myös muita sovelluksia on siitä tehty. Esim. sauna ilmoittaa kun se on valmis tai auton näytölle tulee ilmoitus jos rengaspaineet laskevat alle sallitun rajan. Langaton pääsy internetiin on myös Bluetoothin yksi tärkeimmistä ominaisuuksista. Tämän avulla päästään eroon johdoista. Yhteys voidaan ottaa esim. Bluetooth-tukiaseman kautta. (Kontio, Tervo...2002: 69 - 71.)

7.3 Tiedonsiirto

Bluetooth käyttää taajuutta 2.4GHz. Tämä käytössä oleva taajuusalue on jaettu 1 MHz:n taajuusalueisiin. Kanavia saadaan näin 79 kpl. Tämä edellyttää, että koko taajuusalue on käytössä. Kuten edellä mainittiin, joissakin maissa voi olla vain osa taajuusalueesta käytössä tai siinä on muita rajoitteita. Jos vain osa taajuusalueesta on käytössä, on silloin 23 kanavaa käytössä 79:n sijaan.

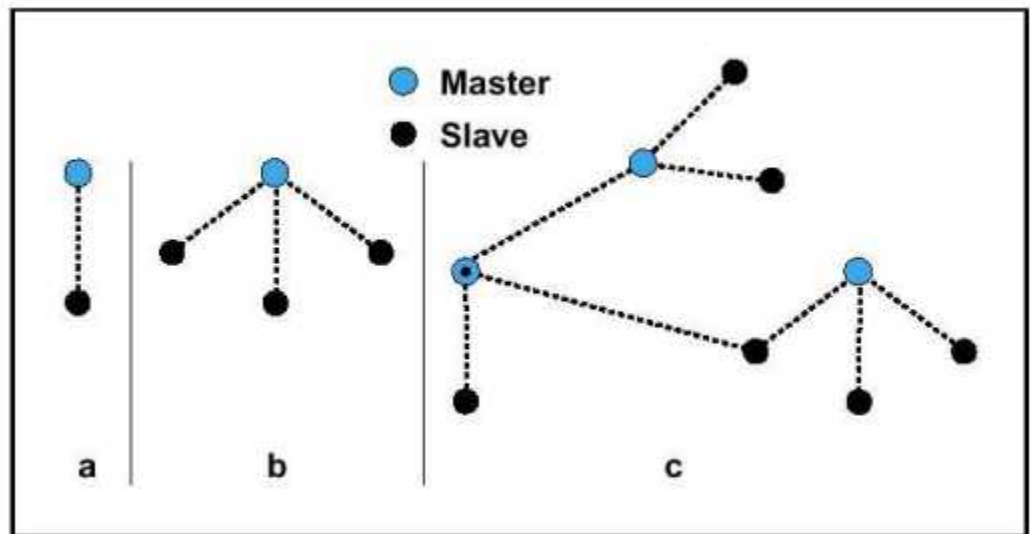
Tieto siirtyy kanavia pitkin paketteina. Paketit lähetetään satunnaisesti valittua kanavaa pitkin aina eri taajuudella. Tätä kutsutaan taajuushyppelyksi. Yhden aikavälin pituus on 625 mikrosekuntia. Taajuushyppely parantaa pakettien perille pääsyä, koska taajuusalueen ollessa häirittynä ainoastaan ne paketit lähetetään uudestaan, jotka eivät päässeet perille.

Tiedonsiirrossa käytetyllä Hajaspektritekniikalla (FHSS, Frequency Hopping Spread Spectrum) tarkoitetaan signaalien hajottamista laajalle taajuusalueelle. Tämä vähentää häiriöiden esiintymistä. (Kontio, Tervo...:71 - 75.)

Bluetooth käyttää tiedon siirtoon kahta erilaista kanavaa: (SCO, Synchronous Connection-Oriented) synkronista ja (ACL, Asynchronous Connection-Less) asynkronista. Synkroninen tiedonsiirto on tarkoitettu äänen siirtoon. Tiedonsiirtonopeus on 434 kb/s. Nopeus on sama molempiin suuntiin. Asynkronista tiedonsiirtoa käytetään datan siirtoon. Nopeus on 723 kb/s ja paluukanava 57 kb/s. (Kontio, Tervo...:76.)

7.4 Verkko

Bluetooth on suunniteltu toimimaan monipuolisena yhteytenä. Point-to-point linkin lisäksi yhteys useaan pisteeseen on myös mahdollista. Point-to-multipointissa tietoa lähettää yksi laite ja sitä ottaa vastaan useampi laite. Picoverkko (Piconet) syntyy, kun usea laite jakaa saman tiedonsiirtokanavan. Tällaisessa picoverkossa voi olla aktiivisena yksi isäntä (Master), joka määräytyy sen mukaan mikä laite on kytkeytynyt ensimmäisenä verkkoon, sekä seitsemän orjalaitetta (Slave). Verkon isäntälaitteen tehtävä on hoitaa taajuussekvenssien määrittäminen. Tämä johtuu siitä, kun laitteet käyttävät taajuushyppelyä. Silloin pitää olla sovittu sekvenssi taajuuksista, joita käytetään pakettien lähettämiseen. Jos taas orjalaite haluaa lähettää tietoa, on sen pyydettävä lupa isännältä. (Kontio, Tervo... :79,80.)



Kuva 7.2 <http://alpha.rrs.at/docs/bluetooth/folien.html> viitattu 20.3.2005

Kuvassa 7.2 on esitelty Bluetoothin yhteystyypit

- A - kohdassa verkossa on vain kaksi laitetta, eli point-to-point
- B - kohta on esimerkki point-to-multipoint tilanteesta
- C - kohta on esimerkki piconetista, jotka ovat yhteydessä toisiinsa piconetoihin. Tätä kutsutaan sirpaleverkoksi (scatternet operation).

7.5 Bluetooth teollisuudessa

Artikkelissa (Karasti 2000.) teollisuuden kannalta Bluetoothin houkuttelevat ominaisuudet ovat halpuus, yleiskäyttöisyys ja pieni koko. Halpuus on hyvin luonnollinen houkutteleva ominaisuus, joka on pelkästään riittävä syy ottaa uusi teknologia käyttöön. Artikkelissa otettiin kantaa myös olosuhteisiin, jotka ovat vaativia teollisuuskäytössä. Niistä mainittiin lämpö, värinä, ja kosteus sekä sähkömoottorit, jotka voivat tuottaa voimakkaita häiriöitä.

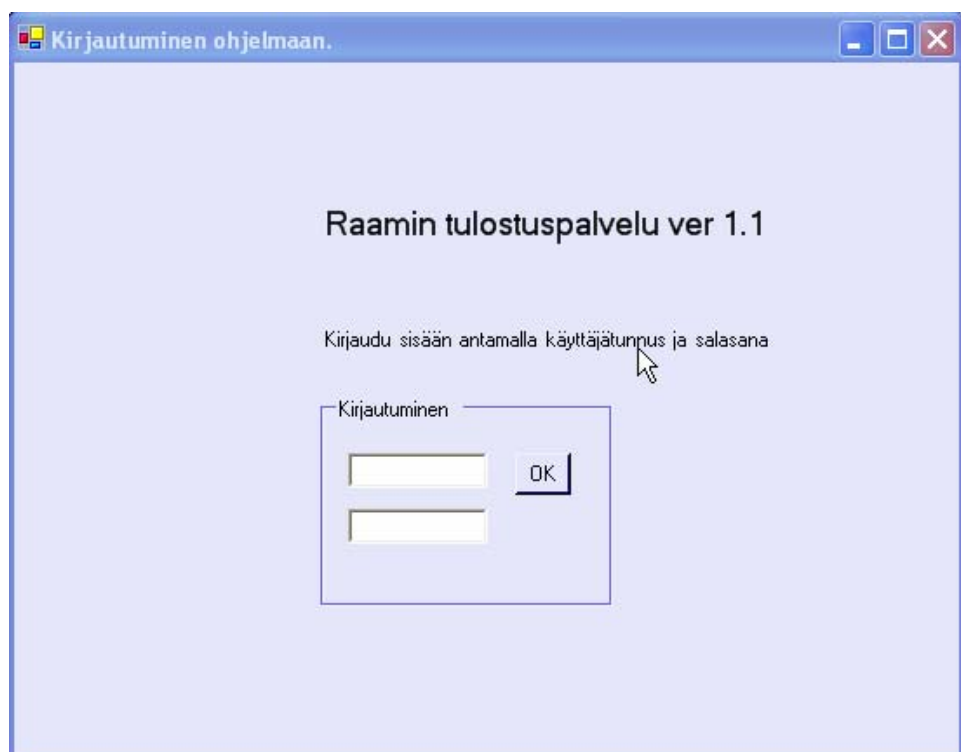
”Sähköiset häiriöt ovat tapauskohtaisia. Ilmassa etenevät sähköiset häiriöt ovat värinää ja kosteutta suurempi ongelma Bluetooth-laitteille. Esimerkiksi suuret sähkökäytöt tuottavat voimakkaita sähköhäiriöitä, jotka hankaloittavat radioliikennettä. Teollisuussovelluksissa tullaankin edellyttämään, ettei Bluetooth-laite aiheuta tiettyä tasoa suurempaa häiriötä ja että se vastaavasti sietää hyvin häiriöitä. ISM-kaistan tehonrajoitus taas aiheuttaa sen, että kohtuullisetkin sähköhäiriöt tukkivat radiokanavan niin, ettei radioliikenne ole enää mahdollista.” (Karasti 2000.)

8 Työn sisältö

Ohjelma ja tietokanta valmistuivat Tammikuussa 2005. Ohjelma toimi tietokannan kanssa testiympäristössä asetettujen tavoitteiden mukaisesti. Teollisuusympäristössä sitä testataan ja otetaan käyttöön kesällä 2005.

Seuraavaksi esitellään ohjelman sisältö. Esittely etenee siinä järjestyksessä, kun ohjelma avautuu. Kuvien lisäksi esitellään koodi ohjelman eri osista.

8.1 Aloitusivu



Kuva 8.1. Ohjelmaan kirjautuminen

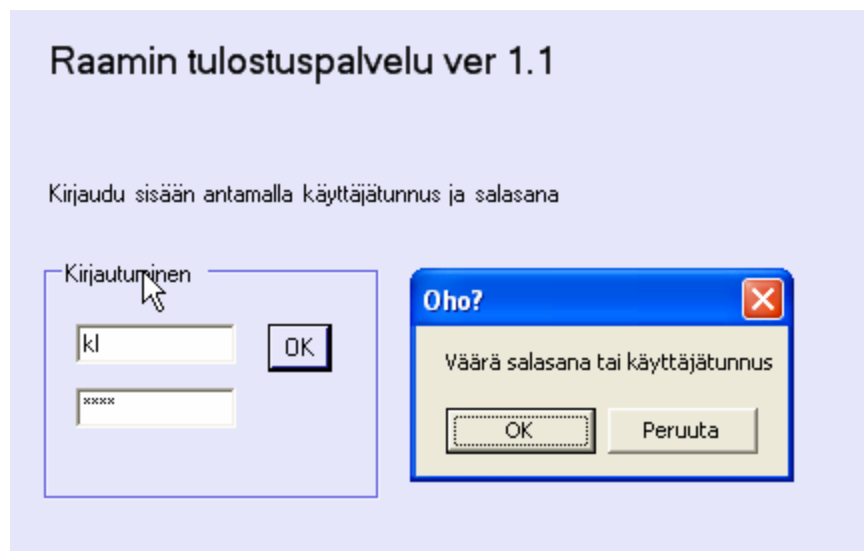
Kuvassa 8.1. on ohjelman aloitusivu, josta kirjaudutaan ohjelmaan. Tunnussana ja käyttäjänimi annetaan ainoastaan niitä tarvitseville. Tunnukset voidaan vaihtaa. Jos käyttäjä antaa väärän salasanan tai käyttäjänimen ei ohjelmaan kirjautuminen onnistu. Ohjelma antaa yrittää kirjautumista niin monta kertaa kuin käyttäjä haluaa.

Koodi: Kirjautuminen ohjelmaan

```
Private Sub cmdok_Click_1(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmdok.Click
```

```
Dim window As New frmaloitus
```

```
If txtkayttaja.Text = "käyttäjätunnus" And txtsalasana.Text = "salasana"  
Then 'Tarkistetaan käyttäjätunnus ja salasana.  
window.ShowDialog()  
WindowsApplication2.frmekasivu.ActiveForm.Hide() 'Jos tunnukset ovat  
oikein, kirjaudutaan ohjelmaan sulkemalla aloitussivu ja avaamalla ohjelman  
pääsivu.  
WindowsApplication2.frmaloitus.ActiveForm.Show()  
Else  
MsgBox("Väärä salasana tai käyttäjätunnus", vbOK, "Oho?") 'Jos tunnukset  
ovat väärin, tulee näytölle siitä ilmoitus.  
txtkayttaja.Text = "" 'Tyhjennetään kentät, jos tunnukset ovat väärin.  
txtsalasana.Text = ""  
End If  
End Sub
```



Kuva 8.2. Väärä salasana tai käyttäjätunnus

Kuvassa 8.2. ohjelma ilmoittaa, jos käyttäjätunnus tai salasana on kirjoitettu väärin. Painamalla 'OK' voi yrittää uudelleen.

8.2 Ohjelman pääsivu

Kuva 8.3. Ohjelman pääsivu

Ohjelman pääsivulta (Kuva 8.3.) löytyvät kaikki ohjelman käyttämiseen tarvittavat kohdat. Ainoastaan tuotteiden päivitys kohta on tehty eri sivulle. Ohjelman käyttöliittymä on suunniteltu siten, että ohjelma olisi helppokäyttöinen. Selkeyden vuoksi sivu on tehty sellaiseksi, että käyttäminen aloitetaan vasemmalta ja siirrytään oikealle.

Ohjelman aloitus

Kuva 8.4. Asiakkaan valinta

Kuvassa 8.4. on ohjelman aloitus. Klikkaamalla "Aloita", "Valitse asiakas" kohta muuttuu aktiiviseksi. "Aloita"-painike muuttuu samalla "harmaaksi", eli sitä ei voi painaa uudestaan.

Jos valitsee asiakkaan, eikä ole painanut ”Aloita”, ohjelma muistuttaa siitä. Ohjelma lisää tyhjän rivin tietokantaan aina kun painetaan ”Aloita”. Samalla kaikki txtkentät tyhjennetään.

Koodi: ”Aloita”-painike

Private Sub cmdaloita_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdaloita.Click

toiminto = 1

txtasiakas.Text = "" 'Kenttien tyhjennys.

txtnro.Text = ""

txtnimi.Text = ""

txt_laatud.Text = ""

txtvarinimi.Text = ""

txtvarinro.Text = ""

txtvari_id.Text = ""

txtera_nro.Text = ""

txtmetrit.Text = ""

txtera_id.Text = ""

txtpvm.Text = DateString 'Päivämäärän laittaminen näytölle.

cmdaloita.Enabled = False 'Aloita painike harmaaksi eli pois käytöstä.

cmdpoista.Enabled = False 'Poista painike harmaaksi eli pois käytöstä.

cmdtallenna.Enabled = True 'Tallenna painike tulee näkyviin, jotta erä voidaan tallentaa.

cmdcancel.Enabled = True 'Cancel painike aktiiviseksi. Tämä keskeyttää ja palauttaa Aloita napin alkutilanteeseen.

cmbajetut.Items.Clear() 'Ajetut kentän tyhjennys.

cmbasiakas.Focus() 'Siirretään Valitse asiakas aktiiviseksi.

cmbasiakas.Enabled = True

End Sub

Asiakkaan valinta ja eränumero

Kun käyttäjä valitsee asiakkaan, ohjelma laskee erälle eränumeron. Eränumero lasketaan hakemalla tietokannasta ko. asiakkaan suurin eränumero ja siihen lisätään 1. Tämä luku tulee näytölle kohtaan eränumero.

Eränumero lasketaan erikseen jokaiselle asiakkaalle. Eränumerot aloitetaan 1:sta. Eränumerot siirretään käytössä olevasta järjestelmästä ohjelmaan siten, että tallennetaan tietokantaan viimeisin eränumero jokaiselle asiakkaalle ja ohjelma jatkaa laskemista siitä.

Koodi: Asiakkaan valinta

```

Private Sub cmbasiakas_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmbasiakas.SelectedIndexChanged
Try
Dim yhteys As OleDbConnection 'Esitellään tietokannan yhteysmuuttuja.
Dim db_komento As New OleDbCommand 'Esitellään tietokannan komento
muuttujat.
Dim db_osoite As String 'Esitellään tietokannan osoite muuttuja.
Dim intenro As Integer 'Esitellään muuttuja, eränumero.
txtasiakas.Text = cmbasiakas.Text 'Txtasiakas kenttään tulee listasta valittu
asiakas.

If cmdaloita.Enabled = True Then
MsgBox("Muista klikata Aloita, ennen ajoa.", vbOK, "Huom.") 'Jos aloita
painiketta ei ole painettu, niin ohjelma muistuttaa siitä.
End If
'Jos asiakas on esim. asiakas1, tulee txtkenttään sama ilman hipsuja jolloin
tiedon tallentaminen onnistuu.
If cmbasiakas.Text = "Valitse asiakas" Then
txtasiakas.Text = ""
ElseIf cmbasiakas.Text = "asiakas1" Then
txtasiakas.Text = "asiakas1" 'Txtasiakas kenttään tulee listasta valittu asiakas.
.....
ElseIf cmbasiakas.Text = "asiakas2" Then
txtasiakas.Text = "asiakas2"
End If
db_osoite = txt_db_osoite.Text Tietokannan osoite, joka kirjoitetaan
txt_db_osoite.text kenttään.
yhteys = New OleDbConnection ("Provider=Microsoft.Jet.OLEDB.4.0;" _
& "DATA SOURCE=" & db_osoite) 'Tietokannan käyttämä ajuri Access
tietokannoissa, sekä tietokannan osoite.
db_komento.CommandText = "SELECT era_nro FROM rekisteri WHERE
asiakas = " & cmbasiakas.Text & "" 'Sql-komento, jolla haetaan eränumero
tietokannasta valitun asiakkaan perusteella.
db_komento.Connection = yhteys
yhteys.Open() 'Avataan yhteys tietokantaan.
Dim db_lue_tieto As OleDb.OleDbDataReader = db_komento.ExecuteReader
'Esitellään tietokannan lukijamuuttuja.
While db_lue_tieto.Read
intenro = db_lue_tieto("era_nro").ToString 'Eränumeron hakeminen.

If intenro.ToString < 1 Then 'Jos eränumero on pienempi kuin 1 laitetaan
eränumeroksi 1.
intenro = 1
txtera_nro.Text = intenro 'Sijoitetaan eränumero näytölle.
Else

```

```

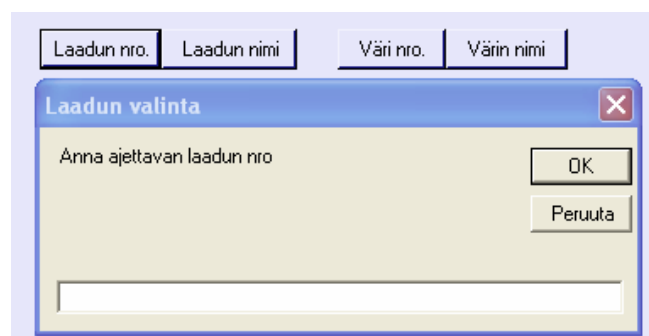
txtera_nro.Text = intenro + 1 'Uuden erän eränumero on viimeksi ajetun erän
eränumero + 1.
End If
End While
yhteys.Close()'Suljetaan yhteys.
Catch err As System.Exception
MsgBox(err.Message)'Antaa virheilmoituksen näytölle.
End Try
End Sub

```

Värin- ja laadun valinta

Seuraavaksi käyttäjä valitsee laadun ja värin. Ne voidaan hakea kummassa järjestyksessä tahansa. Tiedot laaduista ja väreistä sijaitsevat tietokannassa. Tiedot voi hakea joko numeron tai nimen perusteella. Käyttäjä painaa sitä painiketta, jonka perusteella hän haluaa etsiä tiedon.

Tässä esitellään vain laadun hakeminen, koska värin hakeminen tapahtuu aivan samalla tavalla.



Kuva 8.5. Laadun valinta

Kuvassa 8.5. on esimerkki Laadun valinnasta. Käyttäjä on jo klikannut "Laadun nro"-painiketta, koska hän haluaa hakea tiedon laadusta numeron perusteella. Samalla tavalla valitaan myös väri. Jos tietokannasta löytyy haettava laatu tai väri, tulee niiden tiedot näkyviin txtkenttiin. Jos tietoa ei löydy, ei näytöllä näy mitään. Ohjelma ei myöskään anna mitään virheilmoitusta.

Koodi: Laadun hakeminen.

```

Private Sub cmdetsilaatunumerolla_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmdetsilaatunumerolla.Click
etsi_laatu(nro)'Kutsutaan aliohjelmaa etsimään tietoa.
End Sub

```

Public Function etsi_laadunrolla() As String 'Etsitään laadun numerolla ajettava laatu.

Try

Dim yhteys As OleDbConnection 'Esitellään tietokannan yhteysmuuttuja.

Dim db_komento As New OleDbCommand' Esitellään tietokannan komentomuuttuja.

Dim db_osoite As String 'Esitellään tietokannan osoite.

Dim haelaatu As String 'Esitellään haelaatu muuttuja.

haelaatu = InputBox("Anna ajettavan laadun nro", "Laadun valinta")' Laadun numero kirjoitetaan input boxiin, jonka perusteella tieto haetaan. Kun haetaan tietoa nimen perusteella, tapahtuu se samalla lailla.

db_osoite = txt_db_osoite.Text 'Tietokannan osoite, joka kirjoitetaan txt_db_osoite.Text- kenttään.

yhteys = New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;" _ & "DATA SOURCE=" & db_osoite)) 'Tietokannan käyttämä ajuri Access tietokannoissa, sekä tietokannan osoite.

*db_komento.CommandText = "SELECT * FROM laatu WHERE asiakas = " & cmbasiakas.Text & " and laatu_nro =" & haelaatu.ToString & ""*

'Tähän ollaan laitettu valmiiksi kysely, jos halutaan että asiakas on myös hakuehto.

*db_komento.CommandText = "SELECT * FROM laatu WHERE laatu_nro =" & haelaatu.ToString & "" 'Sql-komento, jolla haetaan kaikki tieto taulukosta laatu, laatu_nron perusteella.*

db_komento.Connection = yhteys

yhteys.Open()'Avataan yhteys tietokantaan.

Dim db_lue_tieto As OleDb.OleDbDataReader = db_komento.ExecuteReader 'Esitellään tietokannan lukijamuuttuja.

While db_lue_tieto.Read

'Sijoitetaan tieto näytölle.

txt_laatuid.Text = db_lue_tieto("laatu_id").ToString)

txtnimi.Text = db_lue_tieto("laatu_nimi").ToString)

txtnro.Text = db_lue_tieto("laatu_nro").ToString)

txtas.Text = db_lue_tieto("asiakas").ToString)

End While

yhteys.Close()'Suljetaan yhteys.

cmdpoista.Enabled = True

'Jos haussa on tapahtunut virhe.

Catch err As System.Exception

MsgBox(err.Message)

End Try

End Function

Laadun numero
<input type="text" value="101"/>
Laadun nimi
<input type="text" value="Kangas"/>
Laatu_id
<input type="text" value="41"/>
Asiakas
<input type="text"/>

Kuva 8.6. Laadun tiedot.

- Laadun numero = asiakkaan ilmoittaman laadun numero
- Laadun nimi = asiakkaan ilmoittaman laadun nimi
- Laatu_id = Laadun tunniste
- Asiakas = Laadulle yksilöity asiakas

Kuvassa 8.6. on esitelty Laadun sisältämät tiedot. Alimmainen sarake, ”Asiakas” on sitä varten, jos laadulle halutaan laittaa myös ”Asiakas”-tieto. Tieto asiakkaasta ei ole pakollinen.

Kuivauksen ja rullauksen aloittaminen

Siinä vaiheessa, kun ohjelmasta on valittu asiakas, sekä tiedot laadusta ja väristä, voidaan tekstiilin kuivaus ja rullaaminen aloittaa. Kun pakka on rullattu, kirjoitetaan pakan pituus kohtaan ”Pakan pituus” ja painetaan ”Tulosta”. Tällöin pakan pituus tulostuu muiden tietojen kanssa pakkalapulle. Tulostettaessa pakkalappua, pakan pituus sijoitetaan muuttujaan. Tänne sijoitetaan koko erän pakkojen pituudet.

Tulostaminen

Tulostus tulee tapahtumaan silloin, kun koneen käyttäjä näkee pakan sauman raamin alkupäässä ja painaa pakan katkaisu painiketta. Tämä on komento leikkurille, joka sijaitsee raamin loppupäässä.

Sauman mentyä Raamin läpi loppuun, se katkeaa ja uuden pakan rullaaminen alkaa. ”Tulostus”-painiketta painetaan samalla, kun pakan katkaisu painiketta. Tulostaminen tapahtuu heti ja pakkalappu on valmis, kun sauma tulee leikkurille ja valmis pakka tippuu pöydälle paketoitavaksi. Ohjelman pitää lisätä pakan pituuteen se määrä metrejä, jonka pakan sauma kulkee raamin alkupäästä ”Tulosta”-painikkeen painalluksen jälkeen leikkurille.

Kohtaan ”Erän pituus” tulee koko erän yhteispituus. Pituuden ohjelma laskee tulostetuista pakoista. Aina kun ohjelma tulostaa pakkalapun, siirretään ”Pakan pituus” kohtaan ”Erän pituus”. Kun koko erä on ajettu, painetaan ”Tallenna”-painiketta, jolloin ohjelma kysyy tallennetaanko erän tiedot tietokantaan.

Koodi: ”Tulosta”-painikkeen toiminta

```
Private Sub cmdtulostus_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdtulostus.Click
'Tulostuksen ohjaus. tulostaa tiedot kaikista kentistä, yhden kerran.
Dim pit As Integer
pit = CInt(txtpakka.Text)
kokopit = pit + kokopit 'Tämä laskee aina tulostetun pakan pituuden, erän
koko pituuteen.
txtmetrit.Text = kokopit
Try
Dim printdoc As New PrintDocument
AddHandler printdoc.PrintPage, AddressOf Me.printtext
printdoc.Print() 'Aliohjelman kutsu.
Catch err As System.Exception
MsgBox(err.message)
End Try
txtpakka.Clear() 'Tyhjää ”Pakan pituus”- kentän, kun pakkalappu on
tulostettu.
End Sub
```

Painamalla ”Tulosta”-painiketta, tulostuu valmis pakkalappu.

Värjäämö Laikola & Co
Asiakas:
Laatu (nro):
Laatu (nimi):
Väri (nimi):
Väri (nro):
Eränro:
Ajettu: 02-07-2005
Metrit
Reklamaatiot tehtävä 7 vrk:n sisällä,
ainoastaan leikkaamattomista pakoista

Kuva 8.7. Valmis pakkalappu

Koodi: Pakkalappu

*Private Sub **printtext**(ByVal sender As Object, ByVal ev As PrintPageEventArgs)*

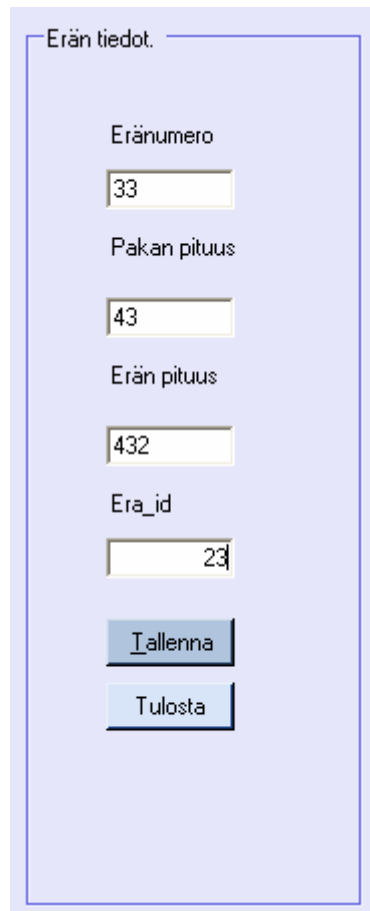
'Pakkalappun sisältämä tieto ja niiden asettelu paperille.

```

ev.Graphics.DrawString("Värjäämö Laikola & Co", New Font("Arial", 11,
FontStyle.Bold), Brushes.Black, 20, 0)
ev.Graphics.DrawString("Asiakas:", New Font("Arial", 8, FontStyle.Bold),
Brushes.Black, 20, 20)
ev.Graphics.DrawString(txtasiakas.Text, New Font("Arial", 8,
FontStyle.Bold), Brushes.Black, 100, 20)
ev.Graphics.DrawString("Laatu (nro):", New Font("Arial", 8, FontStyle.Bold),
Brushes.Black, 20, 40)
ev.Graphics.DrawString(txtnro.Text, New Font("Arial", 8, FontStyle.Bold),
Brushes.Black, 100, 40)
ev.Graphics.DrawString("Laatu (nimi):", New Font("Arial", 8,
FontStyle.Bold), Brushes.Black, 20, 60)
ev.Graphics.DrawString(txtnimi.Text, New Font("Arial", 8, FontStyle.Bold),
Brushes.Black, 100, 60)
ev.Graphics.DrawString("Väri (nimi):", New Font("Arial", 8, FontStyle.Bold),
Brushes.Black, 20, 80)
ev.Graphics.DrawString(txtvarinro.Text, New Font("Arial", 8,
FontStyle.Bold), Brushes.Black, 100, 80)
ev.Graphics.DrawString("Väri (nro):", New Font("Arial", 8, FontStyle.Bold),
Brushes.Black, 20, 100)
ev.Graphics.DrawString(txtvarinimi.Text, New Font("Arial", 8,
FontStyle.Bold), Brushes.Black, 100, 100)
ev.Graphics.DrawString("Eränro:", New Font("Arial", 8, FontStyle.Bold),
Brushes.Black, 20, 120)
ev.Graphics.DrawString(txttera_nro.Text, New Font("Arial", 8,
FontStyle.Bold), Brushes.Black, 100, 120)
ev.Graphics.DrawString("Ajettu:", New Font("Arial", 8, FontStyle.Bold),
Brushes.Black, 20, 140)
ev.Graphics.DrawString(txtpvm.Text, New Font("Arial", 8, FontStyle.Bold),
Brushes.Black, 100, 140)
ev.Graphics.DrawString("Metrit", New Font("Arial", 8, FontStyle.Bold),
Brushes.Black, 20, 160)
ev.Graphics.DrawString(txtpakka.Text, New Font("Arial", 8, FontStyle.Bold),
Brushes.Black, 100, 160)
ev.Graphics.DrawString("Reklamaatiot tehtävä 7 vrk:n sisällä,", New
Font("Arial", 8, FontStyle.Bold), Brushes.Black, 20, 180)
ev.Graphics.DrawString("ainoastaan leikkaamattomista pakoista", New
Font("Arial", 8, FontStyle.Bold), Brushes.Black, 20, 200)
ev.HasMorePages = False

```

Erän tiedot



Erän tiedot.

Eränumero
33

Pakan pituus
43

Erän pituus
432

Era_id
23

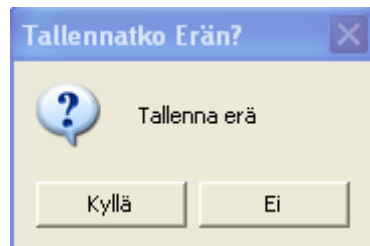
Tallenna

Tulosta

Kuva 8.8. Erän tiedot

- Eränumero = Jokaiselle erälle pitää olla oma eränumero.
- Pakan pituus = Jokaisen pakan pituus tulostuu pakkalapulle.
- Erän pituus = Erän kokonaispituus.
- Era_id = Tunnistekenttä erälle.
- Tallenna = Tästä voidaan tallentaa erä tietokantaan.
- Tulosta = Tulostaa pakkalapun ja siirtää pakan pituuden lukeman ”Erän pituus”-kenttään.

Erän tallentaminen



Kuva 8.9. Tallenna erä

Painettaessa ”Tallenna erä”-painiketta tulee näytölle ikkuna (Kuva 8.9.). ”Tallenna erä”-painiketta painetaan sitten, kun kaikki samaan erään kuuluvat pakat ovat valmiit.

Koodi: ”Tallenna”-painikkeen toiminto

```
Private Sub cmdtallenna_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdtallenna.Click
If MsgBox("Tallenna erä", MsgBoxStyle.Question Or MsgBoxStyle.YesNo,
"Tallennatko Erän?") = MsgBoxResult.No Then
Exit Sub
End If
'Jos vastataan 'Ei' tallenneta, ei tapahdu mitään, ja jos 'Kyllä', niin siirrytään
tallentamaan.
If toiminto = 1 Then
uusi_tieto() 'Jos toiminto = 1, siirrytään uusi_tieto().
End If
```

```
Public Function uusi_tieto() As String 'Uusi tieto.
Try
Dim db_komento As OleDbCommand 'Esitellään tietokannan
komentomuuttujat.
Dim yhteys As OleDbConnection 'Esitellään tietokannan yhteysmuuttuja.
Dim intlid, intvid, streranro, stretr As Integer 'Esitellään kokonaisluku-
muuttujat.
Dim strasiakas, db_osoite As String 'Esitellään tietokannan osoitemuuttuja
Dim db_kirjoita As New OleDbCommand 'Esitellään komentomuuttuja.
Dim pvm As String 'Esitellään päivämäärämuuttuja.
stretr = txtmetrit.Text
```

```
'Näistä kentistä haetaan tallennettava tieto.
streranro = CInt(txtera_nro.Text)
pvm = txtpvm.Text
strasiakas = txtasiakas.Text
intlid = txt_laatuid.Text
intvid = txtvari_id.Text
db_osoite = txt_db_osoite.Text 'Tietokannan osoite löytyy txt_osoite kentästä.
yhteys = New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;" _
& "DATA SOURCE=" & db_osoite) 'Tietokannan käyttämä ajuri Access
tietokannoissa, sekä tietokannan osoite.
db_kirjoita.CommandText = "Insert Into rekisteri " _Tietokantaan
tallentaminen.
```

```
'Tietokannan sisältämät kentät, joihin tieto tallennetaan.
& "(era_metrit, era_nro, pvm, asiakas, laatu_id, vari_id) " _
& "Values ("
```

'Näistä muuttujista tieto tallennetaan tietokantaan.

```
& "" & stremetr & ", " _
& "" & streranro & ", " _
& "" & pvm & ", " _
& "" & strasiakas & ", " _
& "" & intlid & ", " _
& "" & intvid & ")"
```

'Rekisteri kentän tiedot, joihin tallennetaan erät.

db_kirjoita.Connection = yhteys

db_kirjoita.Connection.Open() 'avataan yhteys

db_kirjoita.ExecuteNonQuery()'Tällä voidaan suorittaa muita SQL-komentoja, paitsi kyselyitä.

yhteys.Close() 'Suljetaan yhteys.

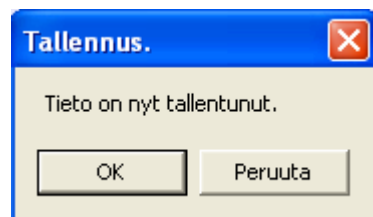
MsgBox("Tieto on nyt tallentunut.", vbOK, "Tallennus.")

Catch err As System.Exception

MsgBox(err.Message)

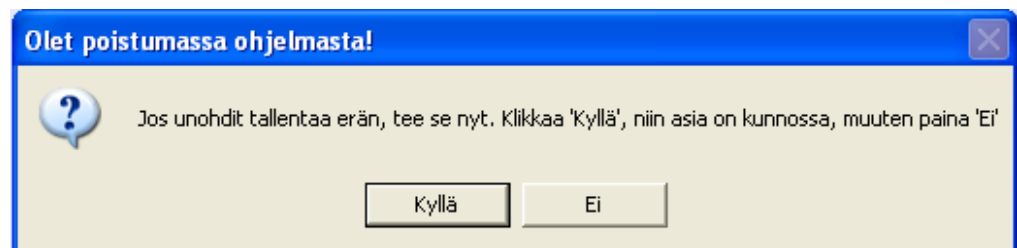
End Try

End Function



Kuva 8.10. Tallennettu

Tietojen tallentaminen ohjelmasta poistuttaessa



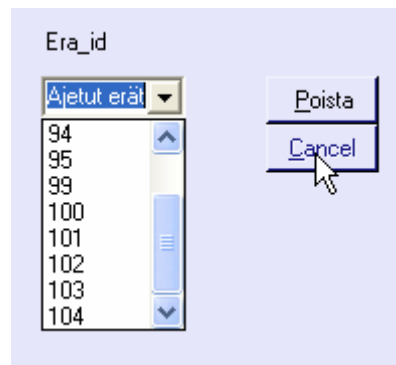
Kuva 8.11. Ohjelmasta poistuminen

Kuvassa 8.11. on ikkuna, joka tulee näytölle kun painat "Sulje ohjelma"-painiketta. Viimeisen erän tiedot voidaan tallentaa, kun poistutaan ohjelmasta.

Koodi: Ohjelmasta poistuminen

```
Private Sub cmdsulje_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdsulje.Click
If MsgBox("Jos unohdit tallentaa erän, tee se nyt. Klikkaa Yes niin asia on
kunnossa, muuten paina 'Ei'", MsgBoxStyle.Question Or MsgBoxStyle.YesNo,
"Olet poistumassa ohjelmasta!") = MsgBoxResult.No Then
End
End If
If toiminto = 1 Then 'Kun ohjelmaa suljetaan, kysytään käyttäjältä onko hän
tallentanut viimeisen erän.
uusi_tieto() ' Jos ei niin hän voi sen tallentaa vielä tässä vaiheessa.
End If
End
End Sub
```

Erien selailu



Kuva 8.12. Erien selailu

Kuvassa 8.12. näkyy, kuinka eriä voi selailla ja poistaa. Kun ohjelmaa käytetään, erien selailu ja poistaminen ei ole mahdollista. Ohjelman käyttäjän painaessa "Cancel"-painiketta, lataa ohjelma tiedot kohtaan "Ajetut erät". Valikkoa selaamalla voi katsoa eriä. Kaikki erän tiedot tulee näytöllä oleviin sarakkeisiin. Kohdasta "Poista", saa poistettua eriä.

Koodi: Ajetut erät

```
Public Function etsi_ajetut() As String 'Haetaan tallennetut tiedot
tietokannasta era_id:n perusteella. Tällä voi selata mitä on ajettu.
```

```
Try
Dim yhteys As OleDbConnection 'Esitellään tietokannan yhteysmuuttuja.
Dim db_komento As New OleDbCommand 'Esitellään
tietokannankomentomuuttuja.
Dim db_osoite As String 'Esitellään tietokannan osoitemuuttuja.
db_osoite = txt_db_osoite.Text ' Tietokannan osoite.
```

```

yhteys = New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;" _
& "DATA SOURCE=" & db_osoite) 'Tietokannan käyttämä ajuri Access
tietokannoissa, sekä tietokannan osoite.
db_komento.CommandText = "SELECT * FROM rekisteri WHERE era_id="
& cmbajetus.Text 'Tietoa haetaan taulukosta rekisteri, era_id:n perusteella.
db_komento.Connection = yhteys
yhteys.Open() 'Avataan yhteys.

```

```

Dim db_lue_tieto As OleDb.OleDbDataReader = db_komento.ExecuteReader
While db_lue_tieto.Read 'Suoritetaan hakukomento.

```

```

'Sijoitetaan haettu tieto näytölle.
txtera_id.Text = db_lue_tieto("era_id").ToString
txtmetrit.Text = db_lue_tieto("era_metrit").ToString
txtera_nro.Text = db_lue_tieto("era_nro").ToString
txtpvm.Text = db_lue_tieto("pvm").ToString
txtasiakas.Text = db_lue_tieto("asiakas").ToString
txtvari_id.Text = db_lue_tieto("vari_id").ToString
txtlaatuid.Text = db_lue_tieto("laatu_id").ToString
End While
yhteys.Close() 'Suljetaan yhteys
cmdpoista.Enabled = True
Catch err As System.Exception
MsgBox(err.Message)
End Try

```

```

Try
Dim yhteys2 As OleDbConnection 'Esitellään tietokannan yhteysmuuttuja.
Dim db_komento2 As New OleDbCommand 'Esitellään tietokannan
komentomuuttuja.
Dim db_osoite As String 'Esitellään tietokannan osoite.
db_osoite = txt_db_osoite.Text 'Osoite kirjoitetaan txt-kenttään.
yhteys2 = New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;" _
& "DATA SOURCE=" & db_osoite) 'Tietokannan käyttämä ajuri Access
tietokannoissa, sekä tietokannan osoite.
db_komento2.CommandText = "SELECT * FROM LAATU WHERE
laatu_id=" & txt_laatu_id.Text
'Komento2, jolla haetaan tieto taulukosta laatu.
db_komento2.Connection = yhteys2
yhteys2.Open()
Dim db_lue_tieto2 As OleDb.OleDbDataReader =
db_komento2.ExecuteReader
While db_lue_tieto2.Read
'Suoritetaan komento ja haetaan tieto.
txtnimi.Text = db_lue_tieto2("laatu_nimi").ToString
txtnro.Text = db_lue_tieto2("laatu_nro").ToString
'Sijoitetaan tieto näytölle
End While

```

```

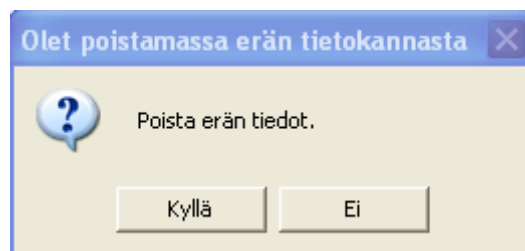
yhteys2.Close()
'Suljetaan yhteys.
cmdpoista.Enabled = True
Catch err As System.Exception
MsgBox(err.Message)
'Mahdolliset virheet tulevat esiin tällä.
End Try

Try
Dim yhteys3 As OleDbConnection 'Esitellään tietokannan yhteysmuuttuja.
Dim db_komento3 As New OleDbCommand 'Esitellään tietokannan
komentomuuttuja.
Dim db_osoite As String 'Esitellään tietokannan osoite.
db_osoite = txt_db_osoite.Text 'Osoite kirjoitetaan näytölle.
yhteys3 = New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;" _
& "DATA SOURCE=" & db_osoite) 'Tietokannan käyttämä ajuri Access
tietokannoissa, sekä tietokannan osoite.
db_komento3.CommandText = "SELECT * FROM vari WHERE vari_id=" &
txtvari_id.Text 'Tiedon hakukomento.
db_komento3.Connection = yhteys3
yhteys3.Open() 'Avataan yhteys.
Dim db_lue_tieto3 As OleDb.OleDbDataReader =
db_komento3.ExecuteReader
While db_lue_tieto3.Read 'Suoritetaan haku.
txtvarinimi.Text = db_lue_tieto3("vari_nimi").ToString()
txtvarinro.Text = db_lue_tieto3("vari_nro").ToString()
'Sijoitetaan tulokset näytölle.
End While
yhteys3.Close()
'Suljetaan yhteys

cmdpoista.Enabled = True
Catch err As System.Exception
MsgBox(err.Message) 'Mahdolliset virheilmoitukset tulevat tällä näkyviin.
End Try
End Function

```

Erien poistaminen



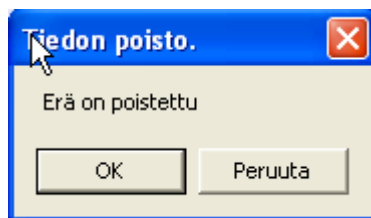
Kuva 8.13. Poista erä

Koodi: Erän poistaminen

```
Public Function poista() As String 'Poistetaan tietokannasta tieto,
ajettu/tallennettu erä.
Try
Dim yhteys As OleDbConnection 'Esitellään tietokannan yhteysmuuttuja.
Dim db_komento As New OleDbCommand 'Esitellään tietokannan
komentomuuttuja.
Dim db_osoite As String
db_osoite = txt_db_osoite.Text 'Tietokannan osoite.
yhteys = New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;" _
& "DATA SOURCE=" & db_osoite) 'Tietokannan käyttämä ajuri Access
tietokannoissa, sekä tietokannan osoite.

db_komento.CommandText = "Delete era_id, era_metrit, era_nro, pvm,
asiakas, laatu_id, vari_if From rekisteri WHERE era_id=" & _
cmbajetut.Text
'Komento, jolla tiedot poistetaan taulukosta.

db_komento.Connection = yhteys
yhteys.Open() 'Avataan yhteys.
db_komento.ExecuteNonQuery()
yhteys.Close()
End Try
End Function
```

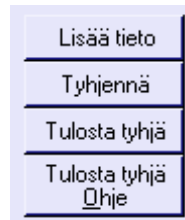


Kuva 8.14. Poistaminen

Koodi: Erä poistettu

```
MsgBox("Erä on poistettu", vbOK, "Tiedon poisto.")
paivita_ajetut() 'Poistettu tieto pitää poistaa myös cbmajetuista, ja se
päivitetään heti.
Catch err As System.Exception
MsgBox(err.Message)
End Try
End Function
```

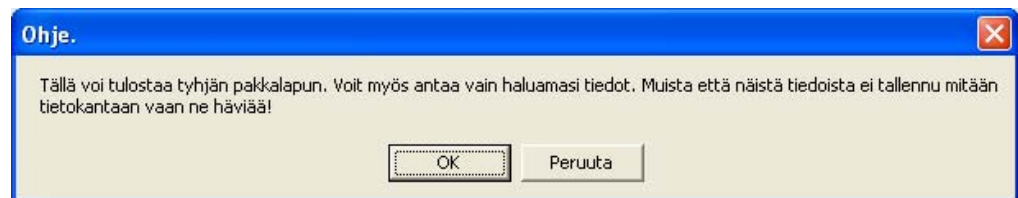
Ohjeet ja painikkeet



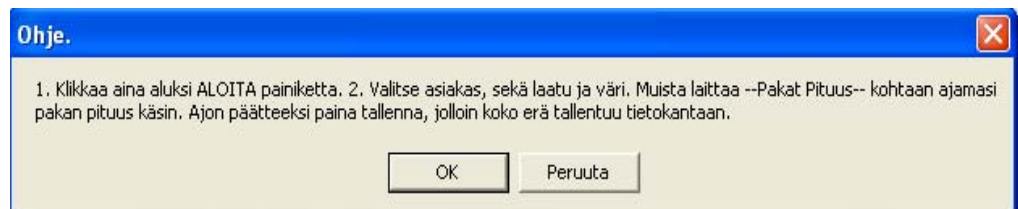
Kuva 8.15 Painikkeet

Painiketekstien selitykset:

- Lisää tieto = Avaa sivun, jossa voidaan päivittää väri- ja laatutietoja.
- Tyhjennä = Tyhjentää kaikki sarakkeet
- Tulosta tyhjä = tulostaa tyhjiä pakkalappuja
- Tulosta tyhjä Ohje = Näyttää ohjeen ja lisätietoa asiaan.



Kuva 8.16. Ohje tyhjän pakkalapun tulostukseen



Kuva 8.17. Ohje ohjelman käyttäjälle

8.3 Väri- ja laatutietojen lisäys ja poistaminen

Sivulla ”Tuotteet”, voidaan tietokantaan lisätä ja poistaa väri- ja laatutietoja. Tällä sivulla voidaan myös selata tietokannassa olevia väri- ja laatutietoja.

Kuva 8.18. Väri- ja laatutiedot

Väri- ja laatutietojen päivitys tapahtuu ohjelmallisesti ja muutokset näkyvät heti tietokannassa. Tämä on nopea tapa päivittää tietokantaa ja lisätyt tiedot voidaan ottaa heti käyttöön.

Tiedon lisäys aloitetaan painamalla ”Uusi”-painiketta. Sillä lisätään tietokantaan tyhjä rivi, jonne uusi tieto tallennetaan. ”Valitse asiakas” muuttuu aktiiviseksi. Laadulle ja värille voidaan tarvittaessa määritellä asiakas. Asiakkaan valinta ei ole pakollista. Tieto asiakkaasta ei ole myöskään hakuehto, kun väri- ja laatutietoja haetaan. ”Asiakas”-tieto tulee ohjelman ”Pääsivulla” näytölle, jos se on erikseen lisätty tietokantaan. Tästä käyttäjä voi päätellä, että kyseinen väri tai laatu kuuluu tietylle asiakkaalle. Kun uusi väri- tai laatutieto on laitettu sarakkeisiin, painetaan ”Tallenna” painiketta, jolloin tieto tallentuu tietokantaan. Tämän jälkeen sarakkeet muuttuvat uudestaan harmaiksi, eli uusia tietoja ei voi lisätä, ennen kuin on painanut ”Uusi”-painiketta.

Kohdasta ”laatu_id” näkee heti, että tieto on tallentunut. Siitä pystyy selaamaan tietokantaa väri- tai laatu-id:n perusteella. Tällä sivulla on myös tiedon haku kohta. Se on samanlainen kuin ohjelman pääsivulla. Sillä voidaan hakea väri- tai laatutietoja tietokannasta, jos halutaan varmistaa löytyykö tieto tietokannasta.

Tietoja lisättäessä on mahdollista lisätä sama tieto kahteen kertaan. Ohjelma ei kommentoi sitä mitenkään, sillä se ei tarkista löytyykö ko. väri tai laatu tietokannasta.

Tämä sen takia, koska samalla numerolla voi olla eri väri tai laatu sekä toisinpäin, jolloin sama laatu tai väri on erinumeroinen eri asiakkaalla.

Koodi: Laadun päivitys

Esittelen vain laadun, koska värin poistaminen ja lisääminen tapahtuu samalla tavalla.

```
Private Sub cmduusilaatu_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmduusilaatu.Click
```

```
    toiminto = 1
```

```
    txtnrolaatu.Text = ""
```

```
    txtnimilaatu.Text = ""
```

```
    txtas_laatu.Text = ""
```

```
    'Tyhjentään kentät, kun ollaan tallentamassa uutta tietoa.
```

```
    cmduusilaatu.Enabled = False 'Kun on painetaan "Uusi_laatu"-painike, se
    muuttuu harmaaksi.
```

```
    cmdpoistalaatu.Enabled = False 'Tallennettaessa tietoa ei "Poista" painike
    ole toiminnassa.
```

```
    cmdtallennalaatu.Enabled = True ' "Tallenna_laatu"-painike aktiiviseksi.
```

```
    cmdcancellaatu.Enabled = True ' "Cancel" keskeyttää tallentaminen.
```

```
    cmblaatuid.Items.Clear() 'Tyhjennetään laatu_id lista
```

```
    cmblaatuid.Enabled = False
```

```
    cmbasiakas_laatu.Focus() 'Asiakas kenttä aktiiviseksi, koska siitä aloitetaan
    uusien tietojen lisääminen.
```

```
    cmbasiakas_laatu.Enabled = True ' Asiakkaat näkyviin, jotta sieltä voidaan
    valita.
```

```
    aktiiv_kentat() 'Laadun nimi- ja numero kentät näkyviin.
```

```
End Sub
```

Koodi esim. Laadun tallentaminen

```
Private Sub cmdtallennalaatu_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdtallennalaatu.Click
```

```
    If MsgBox("Tallenna tieto.", MsgBoxStyle.Question Or
```

```
    MsgBoxStyle.YesNo, "") = MsgBoxResult.No Then
```

```
        'Jos vastaa Kyllä, niin laatu tallentuu ja jos Ei, niin se ei tallennu.
```

```
    Exit Sub
```

```
End If
```

```
    If toiminto = 1 Then
```

```
        uusi_laatu() 'Jos vastasit kyllä, niin tallennetaan tieto.
```

```
    End If
```

```
    cmdtallennalaatu.Enabled = False ' "Tallenna laatu" -painike harmaaksi.
```

```
    cmdcancellaatu.Enabled = False ' "Cancel" -painike harmaaksi.
```

```

cmduusilaatu.Enabled = True' "Uusi Laatu" –painike aktiiviseksi.
txtnimilaatu.Text = "" 'Tyhjennetään kentät tallennuksen jälkeen.
txtnrolaatu.Text = "" 'Tyhjennetään kentät tallennuksen jälkeen.
cmbasiakas_laatu.Focus() 'Asiakas kenttä aktiiviseksi.
cmblaatuid.Enabled = True 'Laatujen selaus id:n avulla aktiiviseksi.
paivita_tuotteet_laatu() 'Päivitetään laatutiedot.
Harmaat_kentat()
End Sub

```

```

Public Function uusi_laatu() As String
Try
Dim db_kirjoita As New OleDbCommand 'Esitellään tietokannan
komentomuuttuja.
Dim yhteys As OleDbConnection 'Esitellään tietokannan yhteysmuuttuja.
Dim strLNimi, strLNro, db_osoite, strASlaatu As String 'Esitellään
tekstimuuttujia.
strLNro = txtnrolaatu.Text 'Muuttujien tiedot sijaitsevat txt-kentissä.
strLNimi = txtnimilaatu.Text
db_osoite = txt_db_osoite.Text
strASlaatu = txtas_laatu.Text
yhteys = New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;" _
& "DATA SOURCE=" & db_osoite) 'Tietokannan käyttämä ajuri Access
tietokannoissa, sekä tietokannan osoite.

db_kirjoita.CommandText = "Insert Into laatu " _ 'Kirjoita komento, jolla
laadun tiedot saadaan tallennettua.
& "(laatu_nimi, laatu_nro, asiakas) " _
& "Values (" _
& "" & strLNimi & ", " _
& "" & strLNro & ", " _
& "" & strASlaatu & ")"
db_kirjoita.Connection = yhteys
db_kirjoita.Connection.Open() 'Avataan yhteys
db_kirjoita.ExecuteNonQuery() 'Suoritetaan komento
yhteys.Close() 'Suljetaan yhteys
MsgBox("Tieto tallennettu.", vbOK, "Tallennus.")
Catch err As System.Exception
MsgBox(err.Message)
End Try
End Function

```

Kuva 8.19. Väritietojen selaus

Koodi: Värien selaus id:n avulla

```
Public Function etsivari() As String
Try
Dim yhteys As OleDbConnection 'Esitellään tietokantamuuttujat.
Dim db_komento As New OleDbCommand
Dim db_osoite As String
db_osoite = txt_db_osoite.Text 'Tietokannan osoite.
yhteys = New OleDbConnection t ("Provider=Microsoft.Jet.OLEDB.4.0;" _
& "DATA SOURCE=" & db_osoite) 'Tietokannan käyttämä ajuri Access
tietokannoissa, sekä tietokannan osoite.

db_komento.CommandText = "SELECT * FROM vari where vari_id=" &
cmbvariid.Text 'Haetaan kaikki tiedot taulukosta vari, jossa
vari_id on käyttäjän valitsema.

db_komento.Connection = yhteys
yhteys.Open() 'Avataan yhteys.
Dim db_lue_tieto As OleDb.OleDbDataReader = db_komento.ExecuteReader
While db_lue_tieto.Read 'Luetaan tieto.
```

```
txtvarinimi.Text = db_lue_tieto("vari_nimi").ToString) 'Sijoitetaan luettu tieto  
näytölle.
```

```
txtvarinro.Text = db_lue_tieto("vari_nro").ToString)
```

```
txtas_vari.Text = db_lue_tieto("asiakas").ToString)
```

```
End While
```

```
yhteys.Close()'Suljetaan yhteys
```

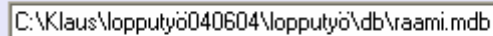
```
btnpoistavari.Enabled = True ' "Poista väri" painike aktiiviseksi.
```

```
Catch err As System.Exception 'Virheen esittäminen.
```

```
MsgBox(err.Message)
```

```
End Try
```

```
End Function
```



C:\Klaus\lopputyö040604\lopputyö\db\raami.mdb

Kuva 8.20. Tietokannan osoite

Kuvassa 8.20. on tietokannan osoite. Osoite kirjoitetaan sen mukaan, missä tietokanta sijaitsee

9 Tietokanta

Tällä hetkellä tietokannassa on käytössä kolme taulukkoa: laatu, vari ja rekisteri. Suunnittelin tietokantaan alun perin oman taulukko myös asiakkaille. Koska asiakkaasta ei kuitenkaan vaadita muuta tietoa, kuin sen nimi, niin päätin laittaa sen rekisteri taulukkoon. Tulin siihen tulokseen että ei kannata tehdä erillistä taulukkoa, jossa on vain yksi tieto.

9.1 Tietokannan tietotyypit ja primary key:t

Tietokannan tietotyypit ovat:

Laatu:

- laatu_id (Primary Key) = laskuri
- laatu_nimi = teksti
- asiakas = teksti
- laatu_nro = teksti

Vari:

- vari_id (Primary Key) = laskuri
- vari_nimi = teksti
- asiakas = teksti
- vari_nro = teksti

Rekisteri:

- era_id (Primry Key) = laskuri
- era_pituus = luku
- era_nro = luku
- pvm = Pvm/klo
- asiakas = teksti
- laatu_id = luku
- vari_id = luku

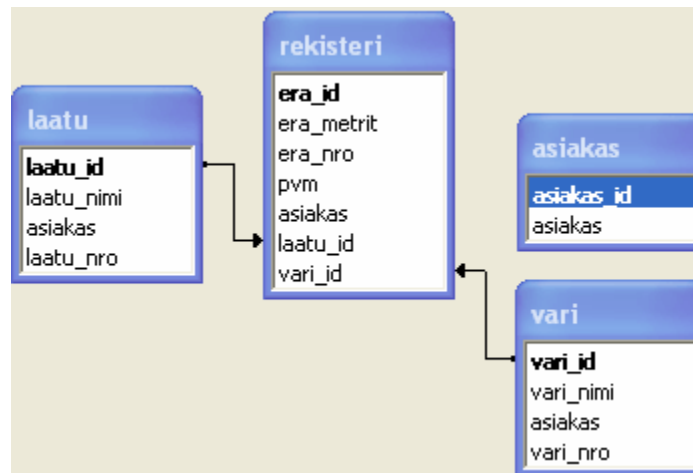
Asiakas

- asiakas_id (Primary key) = laskuri
- asiakas = teksti

Taulukoissa Vari ja Laatu, kenttä Asiakas sallii myös tyhjän arvon. Tämä sen takia, koska värille ja laadulle ei ole pakko laittaa asiakasta. Eri asiakkailta voi olla yhteisiä laatuja ja värejä. Laatu- ja värinumerot voivat olla sekä numeroita että kirjaimia. Tämän takia tietotyyppi on valittu teksti. Esim Laatu: RA64065

9.2 Tietokannan yhteydet

Tietokannan yhteydet ovat Rekisteri taulun laatu_id:stä Laatu taulun laatu_id:een. Samalla tavalla Rekisteri taulu vari_id:stä Vari taulun vari_id:een.



Kuva 9.1. Tietokannan yhteydet

10 Pohdinta

Työn arviointi

Työn tulokseen voin olla tyytyväinen. Työ vastaa sille asetettuja vaatimuksia ja se toimii testiympäristössä. Työn tekemisen aikana aikataulussa pysyminen tuotti hankaluuksia. Työni suurin haaste oli ohjelman tekeminen. Tämä vei aikaa moninkertaisesti enemmän, mitä olin etukäteen suunnitellut, koska jouduin vaihtamaan Visual Basicin versio 6:sta versioon Visual Basic.NET.

Tuotteen arviointi

Tuotteesta, eli ohjelmasta ja tietokannasta, sekä pakkalapuista sain tehtyä sellaiset kuin pitikin. Ohjelma toimii testiympäristössä niin kuin pitääkin. Pakkalaput tulostuvat oikeanlaisina ja oikeilla tiedoilla. Tietokannan osalta sain myös asettamani tavoitteet täytettyä, eli kaikki erät tallentuvat tietokantaan, josta niitä voidaan jälkikäteen tarkastella. Tämän lisäksi tuotteita pystytään lisäämään sekä poistamaan tietokannasta.

Ohjelman vaikutus yrityksen jokapäiväiseen toimintaan selviää lopullisesti ohjelman käyttöönoton jälkeen. Isoimmat vaikutukset tulevat olemaan tiedon varastoinnissa, joka on nyt helpompaa sekä nykyaikaisempaa. Tiedon varastointi on luotettavaa, joka mahdollistaa tiedon tehokkaan jälkikäsiteltävyyden.

Kehittämisehdotuksia

Työn kehittäelyssä riittää haasteita vielä tämänkin jälkeen. Ne liittyvät laitteisiin ja automatiikkaan. Tämän lisäksi tiedolle pitää saada varmempi tallennuspaikka, kuin tietokoneen kiintolevy. Tässä vaiheessa, kun tallennettavaa on paljon, niin oman tiedostopalvelimen laittaminen on hyvinkin kannattavaa. Tiedon hakeminen ja varmuuskopiointi tiedostopalvelimelta on paljon helpompaa, kuin erillisiltä kiintolevyiltä erikseen.

Muita kehitysideoita tähän aiheeseen löytyy myös tietokannan ja ohjelman kehittämisen puolelta. Eräs mahdollisesti käytettävä idea voisi olla yrityksen kotisivuilta pääsy tietokantaan, josta asiakas näkisi reaaliaikaisesti, mitä eriä on jo valmiina haettavaksi.

Lähteet

Internet

Microsoft Corporation 2004, Tuotetietokeskus – Visual Basic.Net [online][viitattu 15.02.2005].

www.microsoft.com/products/info/datasheet.aspx?view=23&pcid=aa13dcca-11d3-4355-a99d-2b86b92caec0

Microsoft Oy 1999, Office 2000 – Microsoft Access[online][viitattu 26.02]

<http://www.microsoft.com/finland/products/office/productinfo/products/access/system.htm>

Karasti, Olavi 2000. Bluetooth teollisuudessa[online][viitattu 31.01.2005].

www.proessori.fi/es00/arkisto/PDF/Bluetooth.PDF

Hänninen, Veijo 2004. Proessori – Uutiset Bluetooth päivittyi kakkosversioon.[online][viitattu 31.01.2005]

www.proessori.fi/uutiset/tulosta.asp?id=46293

Vitikka, Juha 2000. Visual Basic – sovelluskehitin[online][viitattu 22.02.2005]

<http://www.cs.helsinki.fi/u/laine/otv/vitikka.pdf>

Microsoft Corporation, Microsoft Visual Basic.NET Trainer kit [online][viitattu 26.02.2005]

http://www.itinfo.fi/protected.php/visual_basic_net_luku01.pdf?sessid=4b1aa891eabea1195741ec4936b53ec&user=16442&isbn=951-826-703-0

Kuvalähteet

Business Portal – Tpage.com[online][viitattu 20.3.2005].

<http://cool.tpage.com/recommend/bluetooth/img/img1.gif>

<http://alpha.rrs.at/docs/bluetooth/folien.html> [online][viitattu 20.3.2005].

Kirjat

Kontio, Tervo, Jääskeläinen, Arokoski, Vierimaa, Raatikainen ja Köykkä 2002. Mobiiliteknologiat. Helsinki: Edita Prima Oy.

Virenius, Carita. 1997. Access 97. Jyväskylä: Teknolit Oy

Sainio, Arto. 2002. Access 2002. Jyväskylä: Docento Finland Oy

Viescas, John. 2000. Running Microsoft Access 2000. Jyväskylä: Gummerus kirjapaino